



Find: incremental reexecution software test

Documents

Citations

Searching for PHRASE **incremental reexecution software testing tracing**.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web)

CSB DBLP

No documents match Boolean query. Trying non-Boolean relevance query.

500 documents found. Order: relevance to query.

Testing Software for (and with) Data Analysis - John Chambers (1992) (Correct) (1 citation)**Testing Software** for (and with) Data Analysis John M. Chambers**Testing Software** for (and with) Data Analysis John M.

cm.bell-labs.com/cm/ms/departments/sia/doc/93.14.ps

Incremental Testing of Adaptive Software - Keszenheimer, Lieberherr (1994) (Correct) (1 citation)**Incremental Testing of Adaptive Software** Linda M.**Incremental Testing of Adaptive Software** Linda M. Keszenheimer and Karl J. Lieberherr**Incremental Testing of Adaptive Software** Linda M. Keszenheimer and

ftp.ccs.neu.edu/pub/research/demeter/documents/papers/KL94-testing-adaptive.ps

Limitations of Formal Methods and An Approach to Improvement - Liu, Adams (1995) (Correct) (2 citations)large system development if the development is **incremental**. That is, if the high level specification SAsaminami-ku, Hiroshima 731-31 Japan Abstract **Software** development using formal methods is believed toAE Specification Specification Specification **Testing** Program Formalisation Refinement Refinements

www.sel.cs.hiroshima-cu.ac.jp/~liu/APSEC95publish.ps

Foundations of Software Testing: Dependability Theory - Hamlet (1994) (Correct) (3 citations)**Foundations of Software Testing: Dependability Theory** Dick Hamlet**Foundations of Software Testing: Dependability Theory** Dick Hamlet Portland

ftp.cs.pdx.edu/pub/faculty/hamlet/found.ps.Z

A Hybrid Approach to Formal Verification Applied to an ATM... - Clarke, Lee (1996) (Correct)are 1) verification using formal methods and 2) **software testing**. Formal methods use abstract systemis impractical for large systems. In contrast, **testing** allows the search for violations of a property

ftp.cis.upenn.edu/pub/rtg/BKUP/atm.ps.Z

Software Test Coverage and Reliability - Malaiya, Li, Bieman (1996) (Correct) (4 citations)Computer Science Technical Report **Software Test** Coverage and Reliability Yashwant K.Computer Science Technical Report **Software Test** Coverage and Reliability Yashwant K. Malaiya

www.cs.colostate.edu/~ftp/pub/TechReports/1996/tr96-128.ps.Z

A Formal Evaluation of Mutation and Data Flow Based Test... - Mathur, Wong (1994) (Correct)a **test** set is a frequently encountered problem in **software testing**. Data flow and mutation-based **testing**A Formal Evaluation of Mutation and Data Flow Based **Test** Adequacy Criteria Aditya P. Mathur and W. Eric

ftp.cs.purdue.edu/pub/serc/tech-reports/By-School/Purdue/TR133P.PS.Z

On-line Distributed Debugging on Scalable Multiprocessor... - Bemmerl, Wismüller (1994) (Correct) (5 citations)or **reexecution** of small parts of the program (**incremental tracing**" 13]abstract execution" 10]2.2debugging 2. On-line debugging 3. Deterministic **reexecution** 4. Static analysis 2.1 Off-line Debugging An(HLTL) Visualizer VISTOP Load Debugger PATOP **Software** Hardware Hybrid Tasks Monitors Node 0 Node n

www.bode.informatik.tu-muenchen.de/~wismueller/pub/fgs95.ps.gz

Experiments with Data Flow and Mutation Testing - Offutt, Pan, Zhang, Tewary (1994) (Correct) (1 citation)We eliminated redundant **test** cases (by **incrementally** adding **test** cases, and only keeping thosewidely considered to be effective for unit-level **software testing**, but can only be analytically compared**Experiments with Data Flow and Mutation Testing** A. Jefferson Offutt Jie Pan Tong Zhang

www.isse.gmu.edu/techrep/1994/94_105_offutt.ps

The Effect of Correlated Faults on Software Reliability - Wu, Malaiya (1993) (Correct)Science The Effect of Correlated Faults on **Software** Reliability Kang Wu and Yashwant K. MalaiyaThe reliability models often assume random **testing** and statistical independence of faults to keep

www.cs.colostate.edu/~ftp/pub/TechReports/1993/tr-115.ps.Z

An Evaluation Scheme of Software Testing Techniques - Huey-Der Chu (Correct)

1 An Evaluation Scheme of **Software Testing** Techniques Huey-Der Chu Centre for
www.cs.ncl.ac.uk/research/trs/papers/583.ps

Effect of Testing Techniques on Software Reliability Estimates... - Mei-Hwa Chen (1995) (Correct) (2 citations)
Effect of **Testing** Techniques on **Software** Reliability Estimates Obtained Using Time-Domain
Effect of **Testing** Techniques on **Software** Reliability Estimates
hesperus.oboe.com/serc/TechReports/abstracts/authors/.../files/TR108P.PS

GA in program testing - Alander, Mantere, Turunen, Virolainen (Correct)
a pilot we have used a power distribution relay **software**. Keywords: **software** engineering, real time
Chapter 17 GA in program **testing** Jarmo T. Alander, Timo Mantere, Pekka Turunen,
peak.cs.hut.fi/peak/publications/2nwga.ps

Separate Computation of Alias Information for Reuse - Harrold, Rothermel (1996) (Correct) (8 citations)
also provide a way to analyze large systems **incrementally**. 1 Introduction Many **software testing** and
data flow information is useful for many **software testing** and analysis techniques, including data
data flow information is useful for many **software testing** and analysis techniques, including data flow
www.cis.ohio-state.edu/~harrold/research/webpapers/tse96-alias.ps

Some Conservative Stopping Rules for the Operational Testing.. - Bev Littlewood (1997) (Correct) (13 citations)
for the operational **testing** of safetycritical **software** Bev Littlewood David Wright Centre for **Software**
conservative stopping rules for the operational **testing** of safetycritical **software** Bev Littlewood David
seine.pegasus.esprit.ec.org/deva/trs/.../papers/04.ps

NASA Langley's Research and Technology-Transfer.. - Butler, Caldwell, ... (1995) (Correct) (8 citations)
as key areas of research for future avionics **software** and ultrareliable electronics systems [8]1.1
Even after the most thorough and rigorous **testing** some bugs remain. We can never **test** all threads
landing gear due to a design flaw. The problem was **traced** to a timing change in the **software** that had
techreports.larc.nasa.gov/pub/techreports/larc/95/NASA-ieee-95-compassrwb.ps.Z

Automatic Test Generation using Checkpoint Encoding and.. - Yin, Lebne-Dengely.. (1997) (Correct) (2 citations)
Implications of the observations for larger **software** systems are noted. Overall, antirandom **testing**
Computer Science Technical Report Automatic **Test** Generation using Checkpoint Encoding and
www.cs.colostate.edu/~fipub/TechReports/1997/tr97-116.ps.Z

Workload Characterization Using Lightweight System Call.. - Burton, Kelly (1998) (Correct)
Using Lightweight System Call **Tracing** And **Reexecution** Ariel N. Burton And Paul H. J. Kelly
[10] J. M. Mellor-Crummey and T. J. LeBlanc. A **software** instruction counter. In Proc 3 rd
analytical models, simulation, and perhaps also **test** hardware. Such **traces** could also be used for
www.doc.ic.ac.uk/~phjk/Publications/WkidChctrsnUsngTracesReex.ps.gz

On the Estimation of Reliability of a Software System.. - Krishnamurthy, Mathur (1997) (Correct) (7 citations)
On the Estimation of Reliability of a **Software** System Using Reliabilities of its Components
on the sequence of components executed for each **test** input. Path reliability estimates are averaged
described here. Definition 3 The component **trace** of a program P for a given **test** case t is the
hesperus.oboe.com/serc/TechReports/abstracts/authors/.../files/TR172P.PS

Test Selection for Object-Oriented Software Based on.. - Peraire, Barbey, Buchs (Correct) (1 citation)
Test Selection for Object-Oriented **Software** Based on Formal Specifications Ccile Praire
Test Selection for Object-Oriented **Software** Based on
seine.pegasus.esprit.ec.org/deva/trs/.../papers/4C.ps

First 20 documents Next 20

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - Copyright NEC and IST

BEST AVAILABLE COPY



Find: incremental reexecution software test

Documents

Citations

Searching for PHRASE incremental reexecution software testing tracing.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web)

CSB DBLP

No documents match Boolean query. Trying non-Boolean relevance query.

500 documents found. Order: relevance to query.

Testing Real-Time Constraints in a Process Algebraic Setting - Clarke, Lee (1995) (Correct) (4 citations)
over modern digital control systems when the **software** component is taken into account. Whereas the
Testing Real-Time Constraints in a Process Algebraic
ftp.cis.upenn.edu/pub/rig/Paper/Full_Postscript/icse95.ps.Z

Reverse Engineering: Resolving Conflicts between Expected... - Ornburn, Rugaber (1992) (Correct) (4 citations)
Resolving Conflicts between Expected and Actual **Software** Designs Stephen B. Ornburn and Spencer Rugaber
extended. Codelevel analysis, including both the **testing** of expectations and code restructuring, is
www.cc.gatech.edu/reverse/repository/conflicts.ps

Performance Prediction of Paging Workloads Using Lightweight... - Burton, Kelly (2003) (Correct)
to rerunning **traces**: **trace** replay and **trace reexecution**: System call **trace** replay 1 Ariel Burton
on a loosely-coupled distributed system. **Software-Practice and Experience**, 25(10)1117-1140,
times between system calls. The system under **test** is exercised by re-issuing the system calls. The
www.doc.ic.ac.uk/~phjk/Publications/PagingWorkloads-IPDPS-PMEO-2003.ps.gz

Evaluating Empirical Models for the Detection of High-Risk... - Lanubile, Visaggio (1995) (Correct)
20th Annual **Software Engineering Workshop**, November 29-30, 1995,
and main programs. Each group of component was **tested** by independent student teams of an advanced
seldi.uniba.it/pub/papers/sew95.ps

Scalable Monte Carlo Image Synthesis - Heinrich, Arvo (1997) (Correct) (2 citations)
to a given pixel. Pixels of the image are **incrementally** refined by the loop shown below, in which
solutions of the rendering equation. The **software** implementation uses a diffusive load balancing
in the near future. The implementation has been **tested** on a variety of uniprocessors, an Ethernet
www.cs.caltech.edu/~arvo/papers/SMCIS.ps.gz

A Review of Post-Factum Software Integration Methods - Isazadeh, MacEwen, Malton (1995) (Correct)
the original system's functionality. 3. **Incremental Engineering**: A **software** system can be
A Review of Post-Factum **Software** Integration Methods Ayaz Isazadeh Glenn H.
www.qucis.queensu.ca/Department/TechReports/Reports/1995-389.ps

Particle Tracing Algorithms for 3D Curvilinear Grids - Sadarjoen, van Walsum, Hin.. (1994) (Correct) (7 citations)
methods We can distinguish between global and **incremental** point location. In global point location, a
modelling fluid flows. Increasingly sophisticated **software** is being developed to simulate interesting flow
and **tracing** in physical space. Accuracy and speed **tests** are performed for both types of algorithms. From
www.wg.twi.tudelft.nl/~ari/papers/dag94pap.ps.gz

An Integrated Test Environment for the National Defense... - Huey-Der Chu (Correct)
An essential component for developing quality **software** is **software testing**. This paper presents an
kielder.nci.ac.uk/~n4521677/ps/ndmc.ps

An Integrated Test Environment For Distributed Applications - Huey-Der Chu, Dobson (1997) (Correct)
Huey-Der Chu and John E Dobson Centre for **Software** Reliability, Department of Computing Science
kielder.nci.ac.uk/~n4521677/ps/qwe97p.ps

Reproducing Inter-Process Synchronization for Performance... - Burton, Kelly (Correct)
two modes of rerunning **traces**: **trace** replay and **reexecution**. These are described below. **Trace** replay In
reference **tracing**: Implementation and experience. **Software|Practice and Experience**, 26(8)705-736, June
call service times achieved by the system under **test**. **Trace reexecution** In some applications, spinning
www-ala.doc.ic.ac.uk/~phjk/Publications/ReproducingInterProcessSyncUKPEW99.ps.gz

A Statistics-Based Framework For Automated Software Testing - Huey-Der Chu, Dobson (1996) (Correct)
1 A Statistics-Based Framework For Automated **Software Testing** Huey-Der Chu, John Dobson Department Of
kielder.nci.ac.uk/~n4521677/ps/sfast-qw96.ps

An Experiential Approach To Incorporating Software Testing.. - Edward Jones Edward (Correct)
research dimensions of **software testing** in an **incremental** fashion, focusing on a few fundamental
F3d-7 An Experiential Approach To Incorporating **Software Testing** Into The Computer Science Curriculum
An Experiential Approach To Incorporating **Software Testing** Into The Computer Science Curriculum Edward L.
lie.engrng.pitt.edu/lie2001/papers/1385.pdf

Scalable Photorealistic Rendering of Complex Scenes - Heirich, Arvo (1996) (Correct)
to a given pixel. Pixels of the image are **incrementally** refined by the loop shown below, in which
reflective surfaces. In this paper we describe **software** implementation methods and algorithms that
in the near future. The implementation has been **tested** on a variety of uniprocessors, an Ethernet
www.cs.caltech.edu/~arvo/papers/Scalable.ps

Bounding Volume Construction using Point Clouds - Stürzlinger (Correct)
They permit to decide quickly if a more exact **test** is likely to succeed or not. The exact **test** is
Two of the most prominent applications are ray **tracing** and collision detection. This paper presents a
later processing. Results are shown for the ray **tracing** of cyclic CSG-graphs used to render plants and
www.cs.yorku.ca/~wolfgang/papers/pcloud.ps.gz

SITE: A Statistics-based Integrated Test Environment - Huey-Der Chu, Dobson (Correct)
Test Environment Huey-Der Chu Centre for **Software Reliability**, Dept. of Computing Science Bedson
dcs.www.ncl.ac.uk/events/anniversaries/40th/webbook/trs/lists/.papers/584.ps

Incremental Methods for Formal Verification and Logic Synthesis - Swamy (1996) (Correct) (3 citations)
Incremental Methods for Formal Verification and Logic
www-cad.eecs.berkeley.edu/~gms/pub/gmsPhdThesis.ps.Z

Three-Dimensional Computation Visualization - John Stasko (1992) (Correct) (20 citations)
Nevertheless, our system does provide smooth, **incremental** animation effects. **Software** Visualization
program visualization, algorithm information, **software** understanding 1 Introduction For many years
traditional program understanding methods such as **tracing** and debugging. The terms algorithm
ftp.cc.gatech.edu/pub/gvu/tech-reports/92-20.ps.Z

Fostering Theoretical, Empirical and Tool Specific - Barbosa, Adriano (Correct)
Empirical and Tool Specific Knowledge in a **Software Testing** Learning Scenario Ellen F. Barbosa 1
and Tool Specific Knowledge in a **Software Testing** Learning Scenario Ellen F. Barbosa 1
www.dca.fee.unicamp.br/projects/sapiens/calrn/References/Papers/ICECE00/SoftTestScenario/SoftTestScenario.pdf

A Theory of Behaviour Observation in Software Testing - Zhu (1999) (Correct)
Behaviour Observation in **Software Testing** 20/08/99 -1 -A Theory of Behaviour
www.cms.brookes.ac.uk/techreports/cms-tr-99-05.ps.gz

A Methodology to Implement Gourlay's Mathematical Framework.. - Methodology To (1993) (Correct)
This paper presents a methodology to apply a **software testing** theory. The presentation of the
Mathematical Framework for the Investigation of **Testing** June 16, 1993 1 A Methodology to Implement
www.ics.uci.edu/~artreyes/papers/courses/ics280-1993-w-term-paper-implementing-testing-theory.ps.Z

[Documents 21 to 40](#) [Previous 20](#) [Next 20](#)

Try your query at: [Google \(CiteSeer\)](#) [Google \(Web\)](#) [CSB](#) [DBLP](#)

CiteSeer - Copyright NEC and IST

REST AVAILABLE COPY